

Expression Tree

Expression Tree คือ Binary Tree ที่ใช้เก็บนิพจน์ทางคณิตศาสตร์ โหนดของ Expression Tree ประกอบด้วยข้อมูล 2 แบบ คือ

1. Operand คือ ตัวแปร หรือค่าคงที่ที่ถูกดำเนินการทางคณิตศาสตร์ เช่น a, b, x, y หรือ 1, 2, 10, 50 เป็นต้น โดย Operand จะถูกเก็บไว้ที่โหนดใบ
2. Operator คือ เครื่องหมายที่ใช้คำนวณ เช่น +, -, *, / เป็นต้น โดย Operator จะถูกเก็บไว้ที่โหนดภายใน

ข้อสังเกต Sub Tree เป็นนิพจน์ย่อยที่มี Root Node หรือ Parent Node เป็น Operator เสมอ ตัวอย่างดังรูป

จากรูป นิพจน์ทางคณิตศาสตร์ คือ $a * (b + c) - d$

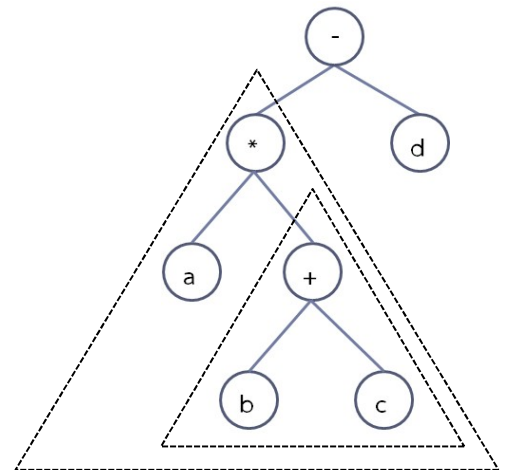
นิพจน์ย่อย คือ $b + c$ และ $a * (b + c)$ (ในกรอบสามเหลี่ยม)

จงเขียนโปรแกรมเพื่อหาผลลัพธ์ของนิพจน์ทางคณิตศาสตร์จาก Expression Tree ที่กำหนดให้

Input : บรรทัดที่ 1 คือ ขนาดของอาร์เรย์ที่ใช้เก็บ Expression Tree
บรรทัดที่สอง คือ ข้อมูลแต่ละตัวจากอาร์เรย์ (เว้นวรรคข้อมูลแต่ละตัว) โดยในที่นี้กำหนดให้

- Operator ประกอบด้วยเครื่องหมาย +, -, * และ / เท่านั้น
- Operand คือ จำนวนเต็มบวก
- หากข้อมูลมีค่าเท่ากับ -1 หมายถึงไม่มีโหนด ณ ตำแหน่งนั้น จึงไม่ต้องนำมาใช้คำนวณ

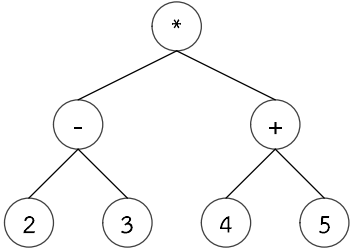
ตัวอย่างเช่น {-, *, d, a, +, -1, -1, -1, -1, b, c} คือ อาร์เรย์ที่ใช้เก็บ Expression Tree ของรูปข้างต้น



Output : ผลลัพธ์ที่ได้จากการคำนวณของ Expression Tree (ทศนิยม 2 ตำแหน่ง)

Sample :

Input	Output	Note
11 + - 5 2 * -1 -1 -1 -1 3 4	-5	

7 * - + 2 3 4 5	-9	
15 + 2 * -1 -1 3 - -1 -1 -1 -1 -1 5 4	5	